Electronic Notes in Theoretical Computer Science 7 (1997)
URL: http://www.elsevier.nl/locate/entcs/volume7.html 20 pages

Absolute versus Relative Time in Process Algebras

Flavio Corradini [1]

Dipartimento di Matematica Pura ed Applicata
Università degli Studi di L'Aquila
L'Aquila, Italia

Abstract

Timed process algebras are useful tools for the specification and verification of real time systems. We study the relationships between two of these algebras, cIPA ('Closed Interval Process Algebra') and *TCCS* ('Temporal *CCS*'), which deal with temporal aspects of concurrent systems by following very different interpretations: durational actions versus durationless actions, absolute time versus relative time, timed functional behavior versus time and functional behavior, local clocks versus global clocks. We show that these different choices are not irreconcilable by presenting a simple mapping from cIPA to *TCCS* which preserves the behavioral equivalences over the two timed calculi.

1 Introduction

In the last years, the classic process algebras such as *CCS* [16], *CSP* [13], *ACP* [4] and Π-calculus [17], have been extended in order to take into account, besides the *functional behaviour* of concurrent systems (which actions the concurrent systems can do), also the *temporal aspects* of their executions, as for instance, a measure of the time consumed for system execution. These extensions, known as *Timed Process Algebras* (see for example, [2,3,5,8,9,12,14,18,20,22], differ, mainly, in the way *time* and *time passing* are modelled. There are, indeed, several parameters which have influenced the choice in the literature:

- *Durational or Durationless Actions*: Basic actions can take a fixed time to be performed and time passes only due to the execution of real "programmable" actions. In others approaches basic actions are instantaneous and "time passes between them" due to the explicit execution of actions modelling

[1] This work has been partially founded by EEC within the HCM Project EXPRESS, and by CNR within the project "Specifica ad Alto Livello e Verifica di Sistemi Digitali".

the passing of time and based upon all parallel components of a concurrent system must synchronize.

- *Absolute or Relative Time*: During a system execution, while experimenting over processes, time stamps can be associated to the observed events. If time stamps are referred to the starting time of the system execution, then we will talk about *absolute time* while if they are referred to the time instant of the previous observation then we will talk about *relative time*.
- *Timed Functional Behaviour or Time and Functional Behaviour*: The study of the functional and temporal behaviour of a concurrent system can be done by integrating the two aspects together, or by separating into two orthogonal parts the functional description and the temporal one.
- *Local Clocks or Global Clocks*: There is a unique (global) time notion but, likewise distributed systems, local clocks are associated to the parallel components of a concurrent system. Alternatively, the notion of global time, and hence of a unique global clock, is made explicit.

The main aim of this paper is to study the relationships between two well-known timed process algebras that implement one or the choice among those listed above. In particular, we consider Aceto and Murphy's cIpa ('Closed Interval Process Algebra') and Moller and Tofts's *TCCS* ('Temporal *CCS*') and we show that the two algebras are strictly related.

cIpa is a *CCS*-like language. External to the syntax a *durational function* (Δ) for actions is specified which, given a basic action a, returns the time needed for its execution (a positive natural number $\Delta(a)$). This permits to considering a normal syntax for processes and to include time information only in the transitional semantics which is given in terms of labelled transition systems the states of which, called timed states, are obtained by associating a local clock $n \in \mathbb{N}$ ($\mathbb{N}$ denotes the set of natural numbers) to every parallel component of a cIpa process. The elapsing of a local clock is set dynamically during the execution of the actions by the corresponding component. If an action a is executed by a component P, the value n of the clock of P is increased to $n + \Delta(a)$, whilst the local clocks of those parallel components not involved in the execution of a are unaffected. Hence, if P is idle during a transition, its local clock value cannot increase. In other words, each parallel component is *eager* to perform an executable action (or dually actions are *urgent*): the time value is incremented locally only when the executable action is performed. The transitions of the labelled transition systems are of the form $d \xrightarrow[\delta]{\mu@n} d'$, meaning that timed state d can become timed state d' by starting the execution of an action μ, n time units after the computation began. Moreover action μ takes δ time units to be performed. Thus in cIpa actions are durational, time is absolute, functional behaviour and timed behaviour are integrated into the same framework, and clocks are local.

TCCS is obtained by extending the syntax of *CCS* with temporal prefixes $(t).p$ representing the process that can evolve into process p after exactly t time units of time. Thus t features as a relative time. Basic actions are durationless: process $a.p$ represents a process which can perform an *instantaneous* action a and then behaves like p. Hence the time passes only due to the execution of temporal prefixings $(t).\_$. The (global) time can elapse of n time units, if n time units can pass for all parallel components. The functional behaviour of processes and the temporal one are described by two different labelled transition systems. One of them describes the functional behaviour and is similar to the standard untimed transition system; states are *TCCS* terms and transitions are of the form $p \stackrel{\mu}{\longrightarrow} q$ meaning that process p can become process q by performing an action μ. The other one describes how processes can change by the time passing; here, states are *TCCS* terms and transitions are of the form $p \stackrel{t}{\leadsto} q$ meaning that process p can become process q after $t \in \mathbb{N}$ time units of time. Summarizing, in *TCCS* actions are durationless, time is relative, functional behaviour and timed behaviour are separated, and there is a unique global clock.

We show that cIpa and *TCCS* are strictly related by showing that when "Ill-Timed" [2] traces are removed by the former algebra, then there exists a very simple mapping from cIpa to *TCCS* which preserves (strong bisimilation-based) behavioural equivalences defined over the two timed calculi. This permits to transferring techniques and analytic concepts from one theory to the other.

A slight modification of the mapping allows us to prove a similar result when, instead of observing the starting times and the duration of actions $(d \xrightarrow[\delta]{\mu@n} d')$, we observe the completing time of action execution. This is the line taken in [9] and [11,12]: transitions are of the form $d \stackrel{\mu@n}{\longrightarrow} d'$ meaning that timed process d can become timed process d' by performing an action μ, at completing time n (n time units after the computation began).

The rest of the paper is organized as follows. The next section contains a brief presentation of cIpa and *TCCS*. Section 3 presents a well-timed operational semantics for the former calculus while Section 4 introduces a coding from cIpa to *TCCS*. Section 5 is devoted to concluding remarks and further work.

2 Modeling Absolute and Relative Time in Process Algebras

This section briefly recalls the basic assumptions behind two well-known timed process algebras: cIpa, due to L.Aceto and D. Murphy [1,2], and *TCCS*, due to F.Moller and C.Tofts [18].

[2] cIpa presents the so called "Ill-Timed phenomena" [1,2] that permits performed actions to be observed not necessarily in the order given by time.

2.1 A Theory of Processes with Durational Actions

The Calculus cIPA

The language used in [1,2] is a variant of Milner's *CCS*. Below we report its syntax. As usual, the set of atomic actions is denoted by A, its complementation by $\bar{A}$ and $\tau \notin A \cup \bar{A}$ is the invisible action. $Act = A \cup \bar{A}$ (ranged over by $a, b, \ldots$) is the set of visible actions and $Act_\tau = Act \cup \{\tau\}$ (ranged over by μ) is the set of all actions. Complementation is extended to Act_τ by $\bar{\bar{a}} = a$ and $\bar{\tau} = \tau$. $\mathbb{N}$ denotes the set of natural numbers, while $\mathbb{N}^+$ denotes the positive ones. Process *variables*, used for recursive definitions, are ranged over by x.

The set of processes, denoted by $\mathcal{P}_{AM}$ (ranged over by $P, Q, \ldots$), are the closed (i.e., without free variables) and guarded (i.e., variable x in a rec $x.P$ term can only appear within a $a.\_$ prefix or a wait $n.\_$ prefix) terms generated by the following grammar:

$$P := nil \;\Big|\; a.P \;\Big|\; \text{wait } n.\, P \;\Big|\; P + P \;\Big|\; P|P \;\Big|\; P\backslash C \;\Big|\; P[\Phi] \;\Big|\; x \;\Big|\; \text{rec } x.P$$

where $a \in Act$, $C \subseteq Act$ and $n \in \mathbb{N}^+$. Process nil denotes a terminated process. By prefixing a term P with an action a, we get a process term $a.P$ which can do an action a and then behaves like P. wait $n.\,P$ denotes a process which can internally evolve for $n \in \mathbb{N}^+$ time units and then behaves like process P. $P + Q$ denotes alternative composition of P and Q. $P|Q$, the parallel composition of P and Q, is the process which can perform any interleaving of the actions of P and Q or synchronizations whenever P and Q can perform complementary actions. $p\backslash C$ is a process which behaves like P but actions in C, or their complements, are forbidden. $P[\Phi]$ behaves like P but its actions are relabeled according to a duration preserving ($\Delta(a) = \Delta(\Phi(a))$ for every $a \in Act$) relabeling function Φ. Finally, rec $v.\,P$ is used for recursive definitions. For the sake of simplicity, terminal nil's can be omitted; e.g. $a + b.c$ stands for $a.nil + b.c.nil$.

The cIPA Operational Semantics

The assumptions behind the theory of processes with durational actions proposed in [1,2] are the following:

1) *maximal parallelism:* whenever a new sequential subagent is actived, there is always a processor free, ready to execute it. In other words, there is never the need of serializing parallel computations.
2) *eagerness:* there is no time-passing in between the execution of actions from the same subagent; equivalently actions happen as soon as possible.
3) *static duration:* the amount of time needed for the execution of a particular action is fixed once and for all on the basis of the features of the chosen machine. *Action durational functions* (ranged over by Δ, Γ, ...) $\Delta : A \to \mathbb{N}^+$,

are introduced which associate to each action the positive natural number of time units needed for its execution. The duration $\Delta(a)$ of action $a \in A$ will be assumed to be nonzero [3] and constant over all occurrences of a. Δ is extended to *Act* by defining $\Delta(\bar{a}) = \Delta(a)$.

$\mathcal{P}_{AM}$ is equipped with an *SOS* semantics in terms of labeled transition the states of which are terms of a syntax extending that of agents with a local *clock prefixing* operator, $n \Rightarrow \_$, which records the evolution of different parts of a distributed state.

Definition 2.1 *The states are terms generated by the following syntax:*

$$d ::= n \Rightarrow nil \Bigm| n \Rightarrow a.P \Bigm| n \Rightarrow wait\ n'.P \Bigm| n \Rightarrow rec\ x.P \Bigm| d{+}d \Bigm| d|d \Bigm| d\backslash C \Bigm| d[\Phi]$$

where $P,\ rec\ x.\,P \in \mathcal{P}_{AM}$, $n \in \mathbb{N}$, $n' \in \mathbb{N}^+$ *and* $C \subseteq Act$. *The set of states is denoted by* $\mathcal{S}_{AM}$ *(ranged over by* d_1, d_2 *...).*

In order to define a simple operational semantics the shorthand expression $n \Rightarrow P$ is used to mean that n distributes over the operators, till the sequential components. The equations in Table 1, called *clock distribution equations*, show that a term $n \Rightarrow P$ can be reduced to a canonical state, when interpreting these equations as rewrite rules from left to right. We will use $\equiv$ to denote the least congruence which satisfies the axioms in Table 1. Each transition is

$n \Rightarrow (P \mid Q) = (n \Rightarrow P) \mid (n \Rightarrow Q)$	$n \Rightarrow (P\backslash C) = (n \Rightarrow P)\backslash C$
$n \Rightarrow (P + Q) = (n \Rightarrow P) + (n \Rightarrow Q)$	$n \Rightarrow (P[\Phi]) = (n \Rightarrow P)[\Phi]$

Table 1
Clock Distribution Equations.

of the form $d \xrightarrow[\delta]{\mu @ t} d'$, meaning that starting from timed state d, timed state d' is reachable by the action μ happening at time t with duration δ. This transition relation is given through a set of inference rules listed in Table 2. It is worthwhile observing that these rules are parametric w.r.t. the chosen duration function Δ. Hence, we should write $\rightarrow_\Delta$. For the sake of simplicity, the subscript will always be omitted whenever clear from the context.

A few comments on the rules in Table 2 are now in order. The rule for action prefixing *Act* states that an action a in $n \Rightarrow a.P$ starts its execution at time n and ends at time $n + \Delta(a)$. Similarly for $Wait$. Rules Sum_1 (Sum_2), *Rec*, *Res* and *Rel* for alternative composition, recursion, restriction and relabeling are as expected. Rule Par_1 (Par_2) for the asynchronous execution of an action μ from the left (right) subagent is almost standard. Note that

[3] Hence, processes which can do an infinite number actions in a finite interval of time, also called *Zeno*-processes, cannot be built in cIPA.

$$Act \; \frac{}{n \Rightarrow a.P \xrightarrow[\Delta(a)]{a@n} (n+\Delta(a)) \Rightarrow P}$$

$$Wait \; \frac{}{n \Rightarrow \text{wait } n'.P \xrightarrow[n']{\tau@n} (n+n') \Rightarrow P}$$

$$Sum_1 \; \frac{d_1 \xrightarrow[\delta]{\mu@t} d}{d_1 + d_2 \xrightarrow[\delta]{\mu@t} d} \qquad Sum_2 \; \frac{d_2 \xrightarrow[\delta]{\mu@t} d}{d_1 + d_2 \xrightarrow[\delta]{\mu@t} d}$$

$$Par_1 \; \frac{d_1 \xrightarrow[\delta]{\mu@t} d_1'}{d_1 \mid d_2 \xrightarrow[\delta]{\mu@t} d_1' \mid d_2} \qquad Par_2 \; \frac{d_2 \xrightarrow[\delta]{\mu@t} d_2'}{d_1 \mid d_2 \xrightarrow[\delta]{\mu@t} d_1 \mid d_2'}$$

$$Synch \; \frac{d_1 \xrightarrow[\Delta(a)]{a@n} d_1', \;\; d_2 \xrightarrow[\Delta(a)]{\bar{a}@n} d_2'}{d_1 \mid d_2 \xrightarrow[\Delta(a)]{\tau@n} d_1' \mid d_2'}$$

$$Res \frac{d \xrightarrow[\delta]{\mu@t} d'}{d \backslash C \xrightarrow[\delta]{\mu@t} d' \backslash C} \mu, \bar{\mu} \notin C \qquad Rel \frac{d \xrightarrow[\delta]{\mu@t} d'}{d[\Phi] \xrightarrow[\delta]{\Phi(\mu)@t} d'[\Phi]}$$

$$Rec \; \frac{n \Rightarrow P[\text{rec } x.\, P/x] \xrightarrow[\delta]{\mu@t} d}{n \Rightarrow \text{rec } x.\, P \xrightarrow[\delta]{\mu@t} d}$$

Table 2
The Structural Rules for cIpa.

during a computation only the local clock of the subprocess responsible of the execution is actually increased, while the others local clocks are unaffected. *Synch* rule, dealing with synchronization, says that two partners can synchronize if they are able to perform complementary actions exactly at the same time.

Based on the transition system for $\mathcal{P}_{AM}$ defined in the previous section, an equivalence relation, called *timing equivalence* has been defined by Aceto and Murphy [1,2]. It is based on the branching-bisimulation style of van Glabbek and Weijland [10]. In Section 3 we will provide cIpa with a new operational semantics and processes will be compared according to the classic notion of strong bisimulation. This equivalence allows a smooth comparison between cIpa and *TCCS*.

2.2 A Theory of Timed Processes with Durationless Actions

This section is devoted to presenting *TCCS* [18], a timed process algebra where basic actions are durationless and time is relative.

The Calculus *TCCS*

The set of processes, denoted by $\mathcal{P}_{MT}$ (ranged over by $p, q, \ldots$), are the closed and guarded (i.e., variable x in a rec $x.p$ term can only appear within a $\mu.\_$ prefix or a $(n)\_$ prefix) terms generated by the grammar below:

$$p := 0 \;\Big|\; \mu.p \;\Big|\; (n).p \;\Big|\; \delta.p \;\Big|\; p+p \;\Big|\; p|p \;\Big|\; p\backslash C \;\Big|\; p[\Phi] \;\Big|\; x \;\Big|\; \mathrm{rec}\; x.p$$

where $\mu \in Act_\tau$, $C \subseteq Act$ and $n \in \mathbb{N}^+$.

In this different setting process 0 represents the nil process; here 0 cannot perform any action and cannot proceed through time. $\mu.p$ is the process that can perform an action μ and then evolve into process p. $(n).p$ is the process which will evolve into process p after exactly n units of time. It is worth noting that $(t).p$ does not correspond to the cIpa wait $t.p$. This is because wait $t.p$ is just a timed version of the *CCS* untimed $\tau.p$ and, in fact, it is possible to think of wait $t.p$ as an abbreviation of $(a|\bar{a}.p)\backslash\{a\}$ (where a is not free in p and $\Delta(a) = t$). This immediately leads to distinguish "a choice followed by a wait" and "a wait followed by a choice"; i.e. wait $t.p +$ wait $t.q$ is different from wait $t.(p+q)$ (the timed version of the distinction between $\tau.p+\tau.q$ and $\tau.(p+q)$). *TCCS*, instead, do not allow the "passage of time to decide a choice and, hence, we will have $(n).p + (n).q$ equivalent to $(n).(p+q)$. $\delta.p$ represents a process which can delay arbitrarily long the execution of process p. $p+q$ is the non deterministic choice between p and q. Any initial passage of time must be allowed by both p and q.

Parallel composition ($p|q$), restriction operator ($p\backslash C$), relabeling ($p[\Phi]$) and recursive definitions (rec $x.p$) are also present.

The *TCCS* Operational Semantics

The operational semantics of *TCCS* is given through two transition relations. The first one, $p \xrightarrow{\mu} p'$, is concerned with the execution of basic actions and is the least relation which satisfies the inference rules in Table 3 (it is very similar to the standard *CCS* transition relation).

The other one, $p \overset{t}{\leadsto} p'$, is concerned with the elapsing of time during a process execution and is the least relation which satisfies the inference rules in Table 4.

This latter timed-transition relation holds the following property which can be proven by induction on the depth of transitions.

Proposition 2.2 *Let p be a $\mathcal{P}_{MT}$ term and $s, t \geq 1$. Then: $p \overset{s+t}{\leadsto} q$ if and only if there exists p_1 such that $p \overset{s}{\leadsto} p_1$ and $p_1 \overset{t}{\leadsto} q$.*

A corollary of the above proposition is the following:

Corollary 2.3 *Let p be a $\mathcal{P}_{MT}$ term and $t \geq 2$. Then: $p_1 \overset{t}{\leadsto} p_{t+1}$ if and only if there are p_2, ... , p_t such that $p_1 \overset{1}{\leadsto} \ldots \overset{1}{\leadsto} p_{t+1}$.*

$$BAct \;\frac{}{a.p \xrightarrow{a} p} \qquad BTau \;\frac{}{\tau.p \xrightarrow{\tau} p} \qquad BDel \;\frac{p \xrightarrow{\mu} p'}{\delta.p \xrightarrow{\mu} p'}$$

$$BSum_1 \;\frac{p \xrightarrow{\mu} p'}{p+q \xrightarrow{\mu} p'} \qquad BSum_2 \;\frac{q \xrightarrow{\mu} q'}{p+q \xrightarrow{\mu} q'}$$

$$BPar_1 \;\frac{p \xrightarrow{\mu} p'}{p \mid q \xrightarrow{\mu} p' \mid q} \qquad BPar_2 \;\frac{q \xrightarrow{\mu} q'}{p \mid q \xrightarrow{\mu} p \mid q'}$$

$$BSynch \;\frac{p \xrightarrow{a} p',\;\; q \xrightarrow{\bar{a}} q'}{p \mid q \xrightarrow{\tau} p' \mid q} \qquad BRec \;\frac{p[\mathrm{rec}\ x.\,p/x] \xrightarrow{\mu} p'}{\mathrm{rec}\ x.\,p \xrightarrow{\mu} p'}$$

$$BRes \frac{p \xrightarrow{\mu} p'}{p\backslash C \xrightarrow{\mu} p'\backslash C} \mu,\ \bar{\mu} \notin C \qquad BRel \frac{p \xrightarrow{\mu} p'}{p[\Phi] \xrightarrow{\Phi(\mu)} p'[\Phi]}$$

Table 3
The Structural Rules for Action Execution in *TCCS*.

$$TDel \;\frac{}{\delta.p \overset{t}{\leadsto} \delta.p} \qquad TSum \;\frac{p \overset{t}{\leadsto} p' \text{ and } q \overset{t}{\leadsto} q'}{p+q \overset{t}{\leadsto} p'+q'} \qquad TPar \;\frac{p \overset{t}{\leadsto} p' \text{ and } q \overset{t}{\leadsto} q'}{p \mid q \overset{t}{\leadsto} p' \mid q'}$$

$$TDec \;\frac{}{(s+t).p \overset{s}{\leadsto} (t).p} \qquad TFin \;\frac{}{(t).p \overset{t}{\leadsto} p} \qquad TFur \;\frac{p \overset{s}{\leadsto} p'}{(t).p \overset{s+t}{\leadsto} p'}$$

$$TRec \;\frac{p[\mathrm{rec}\ x.\,p/x] \overset{t}{\leadsto} p'}{\mathrm{rec}\ x.\,p \overset{t}{\leadsto} p'} \qquad TRes \frac{p \overset{t}{\leadsto} p'}{p\backslash C \overset{t}{\leadsto} p'\backslash C} \qquad TBRel \frac{p \overset{t}{\leadsto} p'}{p[\Phi] \overset{t}{\leadsto} p'[\Phi]}$$

Table 4
The Structural Rules for Time Passing in *TCCS*.

Based on transition relations $p \xrightarrow{\mu} p'$ and $p \overset{t}{\leadsto} p'$ a (strong bisimulation-based) relation over $\mathcal{P}_{MT}$ can be defined [18].

Definition 2.4 *($\mathcal{T}$-Equivalence)*

1. *A binary relation $\Re$ over $\mathcal{P}_{MT}$ is a* $\mathcal{T}$-bisimulation *iff for each $(p,q) \in \Re$:*
   a) $p \xrightarrow{\mu} p'$ *implies* $q \xrightarrow{\mu} q'$ *and* $(p', q') \in \Re$*;*
   b) $q \xrightarrow{\mu} q'$ *implies* $p \xrightarrow{\mu} p'$ *and* $(p', q') \in \Re$*;*
   c) $p \overset{t}{\leadsto} p'$ *implies* $q \overset{t}{\leadsto} q'$ *and* $(p', q') \in \Re$*;*
   d) $q \overset{t}{\leadsto} q'$ *implies* $p \overset{t}{\leadsto} p'$ *and* $(p', q') \in \Re$.
2. *Two $\mathcal{P}_{MT}$ processes p and q are* $\mathcal{T}$-equivalent*, denoted $p \sim_{MT} q$, if and only if there exists a $\mathcal{T}$-bisimulation $\Re$ such that $(p,q) \in \Re$.*

The least congruence over $\mathcal{P}_{MT}$ which satisfies the axioms in Table 5, denoted by $\simeq$, will be useful in the rest of this paper:

$(n).(p \mid q) = (n).p \mid (n).q$	$(n).(p \backslash C) = ((n).p) \backslash C$
$(n).(p + q) = (n).p + (n).q$	$(n).(p[\Phi]) = ((n).p)[\Phi]$

Table 5
Time Distribution Equations.

The following lemma can be proven by standard lines:

Lemma 2.5 *Let p and q be $\mathcal{P}_{MT}$ terms such that $p \simeq q$. Then:*

(1) $p \xrightarrow{\mu} p'$ *implies* $q \xrightarrow{\mu} q' \simeq p'$;

(2) $p \overset{t}{\leadsto} p'$ *implies* $q \overset{t}{\leadsto} q' \simeq p'$.

3 A Well-Timed Operational Semantics for cIpa

The theory of processes with durational actions seen in Section 2.1 presents the so called ill-timed phenomena which allows observation traces that do not reflect the order given by time. We can explain better this with an example. Consider $\mathcal{P}_{AM}$ process $a.b \mid c$ where $\Delta(a) = 2$, $\Delta(b) = 1$ and $\Delta(c) = 3$. Intuitively, when the two parallel components are performed by two different processors, an external observer can see the execution of action a from time 0 to 1, the execution of action b from time 1 to 3 and the execution of action c from time 0 to 3. However, the inference rules in Table 2 allow the following trace to be derived out of $0 \Rightarrow a.b \mid 0 \Rightarrow c$:

$$0 \Rightarrow a.b \mid 0 \Rightarrow c \xrightarrow[2]{a@0} 2 \Rightarrow b \mid 0 \Rightarrow c \xrightarrow[1]{b@2} 3 \Rightarrow nil \mid 0 \Rightarrow c \xrightarrow[3]{c@0} 3 \Rightarrow nil \mid 3 \Rightarrow nil.$$

Notice that this trace does not respect the order given by time or, in other words, is Ill-Timed. Indeed, after observing the starting of action a at time 0 and then the starting of action b at time 2, we can observe the starting of action c at time 0 so that, in some sense, "time goes backward".

We now introduce a new operational semantics for cIpa that removes such traces resulting in a Well-Timed operational semantics: every observation trace respect the temporal order.

The new transition relation is obtained by replacing in Table 2 every $d \xrightarrow[\delta]{\mu@t} d'$ with $d \xrightarrow[\delta]{\langle\mu,t\rangle} d'$ and rules Sum_1, Sum_2, Par_1 and Par_2 with the corresponding ones in Table 6. [4]

[4] Rules Sum_1' and Sum_2' also imply that early actions can always disable later conflicting actions.

Sum'_1	$\dfrac{d_1 \xrightarrow[\delta]{\langle\mu,t\rangle} d,\ \neg(d_2 \xrightarrow[\delta']{\langle\gamma,t'\rangle} d' \text{ and } t' < t)}{d_1 + d_2 \xrightarrow[\delta]{\langle\mu,t\rangle} d}$
Sum'_2	$\dfrac{d_2 \xrightarrow[\delta]{\langle\mu,t\rangle} d,\ \neg(d_1 \xrightarrow[\delta']{\langle\gamma,t'\rangle} d' \text{ and } t' < t)}{d_1 + d_2 \xrightarrow[\delta]{\langle\mu,t\rangle} d}$
Par'_1	$\dfrac{d_1 \xrightarrow[\delta]{\langle\mu,t\rangle} d'_1,\ \neg(d_2 \xrightarrow[\delta']{\langle\gamma,t'\rangle} d'_2 \text{ and } t' < t)}{d_1 \mid d_2 \xrightarrow[\delta]{\langle\mu,t\rangle} d'_1 \mid d_2}$
Par'_2	$\dfrac{d_2 \xrightarrow[\delta]{\langle\mu,t\rangle} d'_2,\ \neg(d_1 \xrightarrow[\delta']{\langle\gamma,t'\rangle} d'_1 \text{ and } t' < t)}{d_1 \mid d_2 \xrightarrow[\delta]{\langle\mu,t\rangle} d_1 \mid d'_2}$

Table 6
The New Rules for cIpa.

If we consider process $a.b\,|\,c$, the new transition relation does not allow the ill-timed computation previously presented. Indeed (see proposition below), ill-timed traces cannot be derived by this new transition relation. The following proposition establishes some nice properties of the new transition relation. It can be proven by induction on the depth of the proof of transitions.

Proposition 3.1 *Let $d \in \mathcal{S}_{AM}$. Then:*

(1) $d \xrightarrow[\delta]{\langle\mu,t\rangle} d'$ *implies* $d \xrightarrow[\delta]{\mu@t} d'$ *(*Consistency*)*

(2) $d \xrightarrow[\delta_1]{\langle\mu_1,t_1\rangle} d'$ *and* $d' \xrightarrow[\delta_2]{\langle\mu_2,t_2\rangle} d''$ *imply* $t_1 \leq t_2$ *(*Well-Timedness*)*

(3) $d \xrightarrow[\delta]{\langle\mu,t\rangle} d'$ *implies* $\neg(d \xrightarrow[\delta']{\langle\gamma,t'\rangle} d''$ *with* $t' < t)$ *(*Urgency*)*

On top of the new transition system for cIpa, a strong bisimulation can be defined that equates processes that can perform the same actions at the same time. This equivalence will be shown to be strictly related to the one in [18].

Definition 3.2 *(Timing Equivalence)*

1. *A binary relation $\Re$ over $\mathcal{S}_{AM}$ is a* timed bisimulation *if and only if for each $(d_1, d_2) \in \Re$:*
   a) $d_1 \xrightarrow[\delta]{\langle\mu,t\rangle} d'_1$ *implies* $d_2 \xrightarrow[\delta']{\langle\mu,t\rangle} d'_2$ *and* $(d'_1, d'_2) \in \Re$*;*
   b) $d_2 \xrightarrow[\delta]{\langle\mu,t\rangle} d'_2$ *implies* $d_1 \xrightarrow[\delta']{\langle\mu,t\rangle} d'_1$ *and* $(d'_1, d'_2) \in \Re$.
2. *Two timed states d_1 and d_2 are* timed equivalent*, denoted $d_1 \sim_{AM} d_2$, if and only if there exists a timed bisimulation $\Re$ such that $(d_1, d_2) \in \Re$.*
3. *Two $\mathcal{P}_{AM}$ processes P, Q are* timed equivalent*, denoted $P \sim_{AM} Q$, if and only if $0 \Rightarrow P \sim_{AM} 0 \Rightarrow Q$.*

When Δ is clear from the context, we omit the superscript Δ, as in $d_1 \sim_{AM} d_2$ and $P \sim_{AM} Q$.

4 Translating $\mathcal{P}_{AM}$ processes into $\mathcal{P}_{MT}$ processes

We are now ready to establishing our main results. We prove that cIPA and *TCCS*, different in several respects, are actually strictly related. We first introduce a mapping $\Pi[\![\_]\!]$ from $\mathcal{P}_{AM}$ terms to $\mathcal{P}_{MT}$ terms, and then we prove that two $\mathcal{P}_{AM}$ processes are timed equivalent if and only if their translations are $\mathcal{T}$-equivalent. More in detail, we want to prove the following statement:

Let $P, Q \in \mathcal{P}_{AM}$. Then, $P \sim_{AM} Q$ if and only if $\Pi[\![P]\!] \sim_{MT} \Pi[\![Q]\!]$.

We start by defining our mapping $\Pi[\![\_]\!] : \mathcal{P}_{AM} \to \mathcal{P}_{MT}$. This will be exploited to define a mapping from timed states to $\mathcal{P}_{MT}$ processes.

Definition 4.1 *Let $\Pi[\![\_]\!] : \mathcal{P}_{AM} \to \mathcal{P}_{MT}$ be defined by the following rules:*

$$
\begin{aligned}
\Pi[\![nil]\!] &= \delta.0 & \Pi[\![P + Q]\!] &= \Pi[\![P]\!] + \Pi[\![Q]\!] \\
\Pi[\![a.P]\!] &= a.(\Delta(a)).\Pi[\![P]\!] & \Pi[\![P \mid Q]\!] &= \Pi[\![P]\!] \mid \Pi[\![Q]\!] \\
\Pi[\![wait\ n.\ P]\!] &= \tau.(n).\Pi[\![P]\!] & \Pi[\![P \backslash C]\!] &= \Pi[\![P \backslash C]\!] \\
& & \Pi[\![P[\Phi]]\!] &= \Pi[\![P]\!][\Phi] \\
& & \Pi[\![x]\!] &= x \\
& & \Pi[\![rec\ x.P]\!] &= rec\ x.\Pi[\![P]\!]
\end{aligned}
$$

A few words on the mapping are now in order. $\mathcal{P}_{AM}$ process nil cannot be related to $\mathcal{P}_{MT}$ process 0. Indeed, while nil is (for instance) the unit for parallel composition and non deterministic choice in the durational setting, process 0 behaves like an annihilitor when composed in parallel or in choice with time-guarded processes. This is because 0 cannot perform neither basic actions, nor allows time passing. To give an example, consider $\mathcal{P}_{AM}$ process $nil|a.b$ that, starting at time 0, can perform an action a at time 0 and duration $\Delta(a)$, and then an action b at time $\Delta(a)$ and duration $\Delta(b)$. By mapping nil into 0 we would have $\Pi[\![nil \mid a.b]\!] = 0 \mid a.(\Delta(a)).b.(\Delta(b)).0$ which can only perform an action a and then deadlock.[5] In order to find a $\mathcal{P}_{MT}$ process that behaves like $\mathcal{P}_{AM}$ process nil, we need a process which cannot perform any basic action but allows the passage of time. This is the reason why we have chosen $\delta.0$ (alternatively, we could take rec $x.(1).x$ which has the same wanted behavior rec $x.(1).x \overset{t}{\leadsto}$ rec $x.(1).x$). Consider now $\mathcal{P}_{AM}$ process $a.P$ the execution of which allows to observe an action a at time 0 and duration

[5] This would lead our main statement to fail because, for instance, processes $nil \mid a.b$ and $nil \mid a$ are such that $\Pi[\![nil \mid a.b]\!] \sim_{MT} \Pi[\![nil \mid a]\!]$ while, clearly, $nil \mid a.b \not\sim_{AM} nil \mid a$.

$\Delta(a)$, and then the execution of process P starting at time $\Delta(a)$. We map $a.P$ into a process that can perform a durationless action a and then a relative delay of $\Delta(a)$ time units. After this delay, the execution proceeds with $\Pi[\![P]\!]$; hence, $\Pi[\![a.P]\!] = a.(\Delta(a)).\Pi[\![P]\!]$. Similarly a wait of n time units followed by a process P, wait $n.\ P$, is mapped into a process that can perform a τ instantaneously and then a relative delay of n time units after which the execution proceeds with $\Pi[\![P]\!]$: $\Pi[\![\text{wait } n.\ P]\!] = \tau.(n).\Pi[\![P]\!]$. $\Pi[\![\_]\!]$ is then extended homomorphically over all the others operators.

Unfortunately, our main statement does not hold for the whole $\mathcal{P}_{AM}$ language but for the subset of restriction free $\mathcal{P}_{AM}$ processes. The following proposition shows a pair of processes P and Q such that $\Pi[\![P]\!]$ and $\Pi[\![Q]\!]$ are related by $\mathcal{T}$-equivalence while P and Q are not related by timed equivalence. The reasons for this drawback are similar to those for which process *nil* cannot be mapped to 0.

Proposition 4.2 *There are P and Q, $\mathcal{P}_{AM}$ processes, such that $\Pi[\![P]\!] \sim_{MT} \Pi[\![Q]\!]$ but not $P \sim_{AM} Q$.*

Proof. Consider the pair of processes $P = a\backslash\{a\} \mid b.c$ and $Q = a\backslash\{a\} \mid b$ that are such that $\Pi[\![P]\!] \sim_{MT} \Pi[\![Q]\!]$ but not $P \sim_{AM} Q$. □

In the rest of this paper, due to Proposition 4.2, we will restrict our attention to the subset of restriction free $\mathcal{P}_{AM}$ processes, denoted by $\mathcal{P}_{AM}^{rf}$, and consequently to the corresponding subset of restriction free timed states, denoted by $\mathcal{S}_{AM}^{rf}$.

Because $\sim_{AM}$ is defined over timed states ($\mathcal{S}_{AM}$ terms) we need a way to translate a timed state into a $\mathcal{P}_{MT}$ process. The next definition introduces this further mapping which exploits $\Pi[\![\_]\!]$.

Definition 4.3 *Let $\mathcal{D}[\![\_]\!] : \mathcal{S}_{AM}^{rf} \to \mathcal{P}_{MT}$ be the least relation which satisfies the following rules:*

$$\begin{array}{rll} \mathcal{D}[\![n \Rightarrow nil]\!] = (n).\Pi[\![nil]\!] & n > 0 & \mathcal{D}[\![d_1 + d_2]\!] = \mathcal{D}[\![d_1]\!] + \mathcal{D}[\![d_2]\!] \\ \mathcal{D}[\![0 \Rightarrow nil]\!] = \Pi[\![nil]\!] & & \mathcal{D}[\![d_1 \mid d_2]\!] = \mathcal{D}[\![d_1]\!] \mid \mathcal{D}[\![d_2]\!] \\ \mathcal{D}[\![n \Rightarrow a.P]\!] = (n).\Pi[\![a.P]\!] & n > 0 & \mathcal{D}[\![d[\Phi]]\!] = \mathcal{D}[\![d]\!][\Phi] \\ \mathcal{D}[\![0 \Rightarrow a.P]\!] = \Pi[\![a.P]\!] & & \\ \mathcal{D}[\![n \Rightarrow wait\ n.\ P]\!] = (n).\Pi[\![wait\ n.\ P]\!] & n > 0 & \\ \mathcal{D}[\![0 \Rightarrow wait\ n.\ P]\!] = \Pi[\![wait\ n.\ P]\!] & & \\ \mathcal{D}[\![n \Rightarrow rec\ x.P]\!] = (n).\Pi[\![rec\ x.P]\!] & n > 0 & \\ \mathcal{D}[\![0 \Rightarrow rec\ x.P]\!] = \Pi[\![rec\ x.P]\!] & & \end{array}$$

A simple property of our translation function is the following:

Proposition 4.4 *Let* $P \in \mathcal{P}^{rf}_{AM}$. *Then* $\mathcal{D}[\![0 \Rightarrow P]\!] = \Pi[\![P]\!]$.

The next step in proving our main statement, is to establish a way for relating the states of the transition system associated to a process $d \in \mathcal{S}^{rf}_{AM}$ and those corresponding to its translated version $\mathcal{D}[\![d]\!]$. We will prove the following property: a state d can perform an action μ at time t and duration δ leading to state d' if and only if $\mathcal{D}[\![d]\!]$ can perform a transition consuming t time units, $\mathcal{D}[\![d]\!] \overset{t}{\leadsto} p$, and then an instantaneous μ action leading to q, $p \xrightarrow{\mu} q$. Furthermore, q is the state (up to $\simeq$) obtained by translating d' when all of its local clocks are decreased by t time units. For instance, consider state $2 \Rightarrow b \,|\, 3 \Rightarrow c$ where $\Delta(b) = 1$ and $\Delta(c) = 3$. According to function $\mathcal{D}[\![\_]\!]$, it is $\mathcal{D}[\![2 \Rightarrow b \,|\, 3 \Rightarrow c]\!] = (2).b.(1).\delta.0 \,|\, (3).c.(3).\delta.0$. We have $2 \Rightarrow b \,|\, 3 \Rightarrow c \xrightarrow[1]{\langle b,2\rangle} 3 \Rightarrow nil \,|\, 3 \Rightarrow c$ if and only if $(2).b.(1).\delta.0 \,|\, (3).c.(3).\delta.0 \overset{2}{\leadsto} b.(1).\delta.0 \,|\, (1).c.(3).\delta.0$ and $b.(1).\delta.0 | (1).c.(3).\delta.0 \xrightarrow{b} (1).\delta.0 | (1).c.(3).\delta.0$. Moreover, $(1).\delta.0 | (1).c.(3).\delta.0 = \mathcal{D}[\![3 - 2 \Rightarrow nil \,|\, 3 - 2 \Rightarrow c]\!]$.

In order to formalize this property, new notation and results are needed. We first define a relation that relates a timed state d in $\mathcal{S}^{rf}_{AM}$ to a natural number $n \in \mathbb{N}$ if by decreasing every local clock $m \Rightarrow \_$ appearing in d of n time units we get a timed state (i.e., a term in $\mathcal{S}^{rf}_{AM}$) a part from $n \Rightarrow nil$ which is related to every natural number t (this is because nil, in cIpa, does not stop the time).

Definition 4.5 *Let* $wf \subseteq \mathcal{S}^{rf}_{AM} \times \mathbb{N}$ *be the least relation which satisfies the following inference rules:*

$$\frac{t \in \mathbb{N}}{wf(n \Rightarrow nil, t)} \quad \frac{t \le n}{wf(n \Rightarrow a.P, t)} \quad \frac{t \le n}{wf(n \Rightarrow wait\ n'.\ P, t)} \quad \frac{t \le n}{wf(n \Rightarrow rec\ x.P, t)}$$

$$\frac{wf(d_1, n)\ \ and\ wf(d_2, n)}{wf(d_1 + d_2, n)} \quad \frac{wf(d_1, n)\ \ and\ wf(d_2, n)}{wf(d_1 \,|\, d_2, n)} \quad \frac{wf(d, n)}{wf(d, n)[\Phi]}$$

Then an updating function $up : \mathcal{S}^{rf}_{AM} \times \mathbb{N} \rightarrow \mathcal{S}^{rf}_{AM}$ is needed which given a timed state $d \in \mathcal{S}^{rf}_{AM}$ and a natural number n such that $wf(d, n)$ returns the state d' obtained by decreasing every local clock $m \Rightarrow \_$ appearing in d of n time units.

Definition 4.6 *Let* $up : \mathcal{S}^{rf}_{AM} \times \mathbb{N} \rightarrow \mathcal{S}^{rf}_{AM}$ *be the least relation which satisfies* *the following inference rules:*

$$\begin{aligned}
up(n \Rightarrow nil, t) &= (n-t) \Rightarrow nil && n \geq t \\
up(0 \Rightarrow nil, t) &= 0 \Rightarrow nil && n < t \\
up(n \Rightarrow a.P, t) &= (n-t) \Rightarrow a.P && n \geq t \\
up(n \Rightarrow wait\ n.\ P, t) &= (n-t) \Rightarrow wait\ n.\ P && n \geq t \\
up(n \Rightarrow rec\ x.P, t) &= (n-t) \Rightarrow rec\ x.P && n \geq t \\
up(d_1\ +\ d_2, t) &= up(d_1, t)\ +\ up(d_2, t) \\
up(d_1 \mid d_2, t) &= up(d_1, t) \mid up(d_2, t) \\
up(d[\Phi], t) &= up(d, t)[\Phi]
\end{aligned}$$

Some simple properties, useful to prove our main statements, are now established. The first one shows that $wf(d,t)$ implies $\mathcal{D}[\![up(d,t)]\!]$ is a process in $\mathcal{P}_{MT}$.

Proposition 4.7 *Let $d \in \mathcal{S}^{rf}_{AM}$ and $t \in \mathbb{N}$ such that $wf(d,t)$. Then $\mathcal{D}[\![up(d,t)]\!]$ is a $\mathcal{P}_{MT}$ process.*

Then we can show that $wf(d,t)$ implies $\mathcal{D}[\![d]\!]$ can perform at least t time units.

Proposition 4.8 *Let $d \in \mathcal{S}^{rf}_{AM}$ and $t \in \mathbb{N}$ such that $wf(d,t)$. Then either $t = 0$ and $d = up(d,0)$, or $t > 0$ and $\mathcal{D}[\![d]\!] \overset{t}{\rightsquigarrow} p$.*

The following lemma shows that if $wf(d,t)$ then every action a timed state d can perform is observed at a time greater or equal than t.

Lemma 4.9 *Let $d \in \mathcal{S}^{rf}_{AM}$ and $t \in \mathbb{N}$ such that $wf(d,t)$. Then $d \xrightarrow[\delta']{\langle \mu, t' \rangle} d'$ implies $t' \geq t$.*

A typical property when dealing with translations is the following:

Proposition 4.10 *Let $P \in \mathcal{P}^{rf}_{AM}$ and $t \in \mathbb{N}$. Then $\Pi[\![P[rec\ x.P/x]]\!] = \Pi[\![P]\!][rec\ x.\Pi[\![P]\!]/x]$.*

Others needed properties are:

Lemma 4.11 *Let $P \in \mathcal{P}^{rf}_{AM}$ and $n, t \in \mathbb{N}$. Then*

(1) $n \geq t$ *implies* $wf(n \Rightarrow P, t)$

(2) $\mathcal{D}[\![up(n \Rightarrow P, t)]\!]\ =\ \mathcal{D}[\![(n-t) \Rightarrow P]\!]$

(3) $(n).\Pi[\![P]\!]\ \simeq\ \mathcal{D}[\![up((n+t) \Rightarrow P, t)]\!]$

The following three propositions are fundamental to prove our main statement. The first one relates transitions out of $\mathcal{D}[\![up(d,t)]\!]$ and d. In particular, we show that $\mathcal{D}[\![up(d,t)]\!]$ can perform a basic action μ if and only if timed state d can perform an action μ at time t and a duration δ.

Proposition 4.12 *Let $d \in \mathcal{S}_{AM}^{rf}$ and $t \in \mathbb{N}$ such that $wf(d,t)$. Then:*

(1) $\mathcal{D}[\![up(d,t)]\!] \xrightarrow{\mu} p$ *implies* $d \xrightarrow[\delta]{\langle\mu,t\rangle} d'$ *with* $wf(d',t)$ *and* $p \simeq \mathcal{D}[\![up(d',t)]\!]$;

(2) $d \xrightarrow[\delta]{\langle\mu,t\rangle} d'$ *implies* $\mathcal{D}[\![up(d,t)]\!] \xrightarrow{\mu} p$ *with* $wf(d',t)$ *and* $p \simeq \mathcal{D}[\![up(d',t)]\!]$;

The next proposition shows that whenever a $\mathcal{P}_{MT}$ process $\mathcal{D}[\![up(d,t)]\!] \overset{1}{\rightsquigarrow} p$ then $\mathcal{D}[\![up(d,t)]\!]$ cannot perform any basic action γ and timed state d cannot perform basic actions at a time $t' \le t$.

Proposition 4.13 *Let $d \in \mathcal{S}_{AM}^{rf}$ and $t \in \mathbb{N}$ such that $wf(d,t)$. Then $\mathcal{D}[\![up(d,t)]\!] \overset{1}{\rightsquigarrow} p$ implies $\mathcal{D}[\![up(d,t)]\!] \overset{\gamma}{\not\longrightarrow}$ for every γ, $d \underset{\delta}{\overset{\langle\mu,t'\rangle}{\not\longrightarrow}} d'$ for every μ and $t' \le t$, $wf(d,t+1)$ and $p \simeq \mathcal{D}[\![up(d,t+1)]\!]$.*

The reverse of the previous proposition is the following. If $d \underset{\delta}{\overset{\langle\mu,t'\rangle}{\not\longrightarrow}} d'$ for every μ and $t' \le t$, then $\mathcal{D}[\![up(d,t)]\!]$ can only evolve by performing a $\overset{1}{\rightsquigarrow}$-transition.

Proposition 4.14 *Let $d \in \mathcal{S}_{AM}^{rf}$ and $t \in \mathbb{N}$. If for every $t' \le t$ and $\mu \in Act$ $d \underset{\delta}{\overset{\langle\mu,t'\rangle}{\not\longrightarrow}} d'$, then $wf(d,t)$, $wf(d,t+1)$ and $\mathcal{D}[\![up(d,t)]\!] \overset{1}{\rightsquigarrow} \mathcal{D}[\![up(d,t+1)]\!]$.*

We are now ready to establish the main statement.

Theorem 4.15 *Let P and Q be $\mathcal{P}_{AM}^{rf}$ processes. Then $P \sim_{AM} Q$ if and only if $\Pi[\![P]\!] \sim_{MT} \Pi[\![Q]\!]$.*

Proof. By $P \sim_{AM} Q$ if and only if $0 \Rightarrow P \sim_{AM} 0 \Rightarrow Q$ if and only if $\mathcal{D}[\![0 \Rightarrow P]\!] \sim_{MT} \mathcal{D}[\![0 \Rightarrow Q]\!]$ if and only if (see Proposition 4.4) $\Pi[\![P]\!] \sim_{MT} \Pi[\![Q]\!]$, in order to prove our theorem it is sufficient to prove the more general statement

Let d_1, d_2 be $\mathcal{S}_{AM}^{rf}$ terms. Then $d_1 \sim_{AM} d_2$ iff $\mathcal{D}[\![d_1]\!] \sim_{MT} \mathcal{D}[\![d_2]\!]$.

To prove that $d_1 \sim_{AM} d_2$ implies $\mathcal{D}[\![d_1]\!] \sim_{MT} \mathcal{D}[\![d_2]\!]$ we prove that relation

$$\begin{aligned}\Re_{AM}^{MT} \;=\; \{(p,q) \mid \exists t \in \mathbb{N} \text{ s.t. } & wf(d_1,t),\ wf(d_2,t),\\ & p \simeq \mathcal{D}[\![up(d_1,t)]\!],\ q \simeq \mathcal{D}[\![up(d_2,t)]\!] \text{ and } d_1 \sim_{AM} d_2\}\end{aligned}$$

is a MT-bisimulation. To prove this let $(p,q) \in \Re_{AM}^{MT}$ and assume:

(a) $p \xrightarrow{\mu} p'$. By Lemma 2.5-(1), $\mathcal{D}[\![up(d_1,t)]\!] \xrightarrow{\mu} p_1 \simeq p'$. By Proposition 4.12-(1), $d_1 \xrightarrow[\delta]{\langle\mu,t\rangle} d_1'$ with $wf(d_1',t)$ and $p_1 \simeq \mathcal{D}[\![up(d_1',t)]\!]$. By $d_1 \sim_{AM} d_2$ we have $d_2 \xrightarrow[\delta']{\langle\mu,t\rangle} d_2'$ with $d_1' \sim_{AM} d_2'$. By Proposition 4.12-(2), $\mathcal{D}[\![up(d_2,t)]\!] \xrightarrow{\mu} q_1$ with $wf(d_2',t)$ and $q_1 \simeq \mathcal{D}[\![up(d_2',t)]\!]$. By Lemma 2.5-(1), $q \xrightarrow{\mu} q' \simeq q_1$. By

$p_1 \simeq p'$ and $p_1 \simeq \mathcal{D}[\![up(d'_1,t)]\!]$ we have $p' \simeq \mathcal{D}[\![up(d'_1,t)]\!]$. Moreover by $q' \simeq q_1$ and $q_1 \simeq \mathcal{D}[\![up(d'_2,t)]\!]$ we have $q' \simeq \mathcal{D}[\![up(d'_2,t)]\!]$. Thus $(p',q') \in \Re^{MT}_{AM}$.

(b) $p \overset{n}{\rightsquigarrow} p'$. By Corollary 2.3 $p = p_0 \overset{1}{\rightsquigarrow} p_1 ... p_{n-1} \overset{1}{\rightsquigarrow} p_n = p'$. Consider transition $p_0 \overset{1}{\rightsquigarrow} p_1$ and note that $p_0 \simeq \mathcal{D}[\![up(d_1,t)]\!]$. Then by Proposition 2.5-(2) $\mathcal{D}[\![up(d_1,t)]\!] \overset{1}{\rightsquigarrow} p'_1 \simeq p_1$. By Proposition 4.13 $\mathcal{D}[\![up(d_1,t)]\!] \overset{\gamma}{\not\longrightarrow} p$ for every γ and p, $d_1 \overset{\langle \mu,t' \rangle}{\underset{\delta}{\not\longrightarrow}} d'$ for every μ, d' and $t' \leq t$, $wf(d_1, t+1)$ and $p'_1 \simeq \mathcal{D}[\![up(d_1,t+1)]\!]$. By $p'_1 \simeq p_1$ and $p'_1 \simeq \mathcal{D}[\![up(d_1,t+1)]\!]$ we have $p_1 \simeq \mathcal{D}[\![up(d_1,t+1)]\!]$. By $d_1 \sim_{AM} d_2$ we must have $d_2 \overset{\langle \mu,t' \rangle}{\underset{\delta}{\not\longrightarrow}} d'_2$ for every μ, d'_2 and $t' \leq t$, so that by Proposition 4.14 we have $wf(d_2,t)$, $wf(d_2,t+1)$ and $\mathcal{D}[\![up(d_2,t)]\!] \overset{1}{\rightsquigarrow} \mathcal{D}[\![up(d_2,t+1)]\!]$. By $q_0 = q \simeq \mathcal{D}[\![up(d_2,t)]\!]$ it is $q_0 \overset{1}{\rightsquigarrow} q_1 \simeq \mathcal{D}[\![up(d_2,t+1)]\!]$. Thus $(p_1,q_1) \in \Re^{MT}_{AM}$. By iterating the above reasonings we can find a sequence $q = q_0 \overset{1}{\rightsquigarrow} q_1 ... q_{n-1} \overset{1}{\rightsquigarrow} q_n = q'$ such that $wf(d_1,t+n)$, $wf(d_2,t+n)$, $p' \simeq \mathcal{D}[\![up(d_1,t+n)]\!]$, $q' \simeq \mathcal{D}[\![up(d_2,t+n)]\!]$ and $(p',q') \in \Re^{MT}_{AM}$. By Corollary 2.3 we also have $q \overset{n}{\rightsquigarrow} q'$.

(c) $q \xrightarrow{\mu} q'$ and $q \overset{n}{\rightsquigarrow} q'$ are similar to (a) and (b) respectively.

To prove $\mathcal{D}[\![d_1]\!] \sim_{MT} \mathcal{D}[\![d_2]\!]$ implies $d_1 \sim_{AM} d_2$ we show that relation

$$\begin{aligned} \Re^{AM}_{MT} \;=\; \{(d_1,d_2) \mid \exists t \in \mathbb{N} \text{ s.t. } & wf(d_1,t),\ wf(d_2,t), \\ & p \simeq \mathcal{D}[\![up(d_1,t)]\!],\ q \simeq \mathcal{D}[\![up(d_2,t)]\!] \text{ and } p \sim_{MT} q\} \end{aligned}$$

is a AM-bisimulation. To prove this let $(d_1,d_2) \in \Re^{AM}_{MT}$ and assume:

(a) $d_1 \overset{\langle \mu,n \rangle}{\underset{\delta}{\longrightarrow}} d'_1$. By $(d_1,d_2) \in \Re^{AM}_{MT}$ there exists $t \in \mathbb{N}$ such that $wf(d_1,t)$,ă$wf(d_2,t)$, $p \simeq \mathcal{D}[\![up(d_1,t)]\!]$, $q \simeq \mathcal{D}[\![up(d_2,t)]\!]$ and $p \sim_{MT} q$. First of all observe that by $wf(d_1,t)$, $d_1 \overset{\langle \mu,n \rangle}{\underset{\delta}{\longrightarrow}} d'_1$ and Lemma 4.9 $n \geq t$. Thus we distinguish two cases for n and t.

(1) $t = n$. By Proposition 4.12-(2) we have $\mathcal{D}[\![up(d_1,t)]\!] \xrightarrow{\mu} p_1$ with $wf(d'_1,t)$ and $p_1 \simeq \mathcal{D}[\![up(d'_1,t)]\!]$. By $p_1 \simeq \mathcal{D}[\![up(d'_1,t)]\!]$, $\mathcal{D}[\![up(d_1,t)]\!] \xrightarrow{\mu} p_1$ and Lemma 2.5-(1), $p \xrightarrow{\mu} p' \simeq p_1$. By $p \sim_{MT} q$, $q \xrightarrow{\mu} q_1$ with $p' \sim_{MT} q_1$. By $q \simeq \mathcal{D}[\![up(d_2,t)]\!]$ and 2.5-(1), $\mathcal{D}[\![up(d_2,t)]\!] \xrightarrow{\mu} q' \simeq q_1$. By Proposition 4.12-(1) $d_2 \overset{\langle \mu,t \rangle}{\underset{\delta'}{\longrightarrow}} d'_2$ with $wf(d'_2,t)$ and $q' \simeq \mathcal{D}[\![up(d'_2,t)]\!] \simeq q_1$. It follows that $(d'_1,d'_2) \in \Re^{AM}_{MT}$ because $wf(d'_1,t)$, $wf(d'_2,t)$, $p' \sim_{MT} q_1$ with $p' \simeq \mathcal{D}[\![up(d'_1,t)]\!]$ and $q_1 \simeq \mathcal{D}[\![up(d'_2,t)]\!]$.

(2) $t < n$. By $d_1 \overset{\langle \mu,n \rangle}{\underset{\delta}{\longrightarrow}} d'_1$ and Proposition 3.1-(3) it is $\neg(d_1 \overset{\langle \gamma,t' \rangle}{\underset{\delta'}{\longrightarrow}} d''_1$ with $t' < t)$. Thus by Proposition 4.14, $wf(d_1,t)$, $wf(d_1,t+1)$ and $\mathcal{D}[\![up(d_1,t)]\!] \overset{1}{\rightsquigarrow} \mathcal{D}[\![up(d_1,t+1)]\!]$. Thus by Lemma 2.5-(2) also $p \overset{1}{\rightsquigarrow} p' \simeq \mathcal{D}[\![up(d_1,t+1)]\!]$. By $p \sim_{MT} q$, $q \overset{1}{\rightsquigarrow} q'$ with $p' \sim_{MT} q'$ and by Lemma 2.5-(2), $\mathcal{D}[\![up(d_2,t)]\!] \overset{1}{\rightsquigarrow} q_1 \simeq q'$. By Proposition 4.13 we have $wf(d_2,t+1)$ and $q_1 \simeq \mathcal{D}[\![up(d_2,t+$

1)⟧. We are, now, in the same hypothesis of the theorem where, however, t is increased by 1 time unit. A simple inductive reasoning is sufficient to prove that item (1) will be reached in a finite number of steps. Thus we can assume $d_2 \xrightarrow[\delta']{\langle\mu,t\rangle} d_2'$ with $(d_1', d_2') \in \Re^{AM}_{MT}$.

(b) $d_2 \xrightarrow[\delta]{\langle\mu,n\rangle} d_2'$ is similar to the previous case.

$\square$

5 Further Remarks and Further Work

In this paper we have shown that *TCCS* is at least as much expressive as cIPA by showing that there exists a simple mapping from the latter timed process algebra to the former one which preserves strong bisimulation equivalence over the two calculi. This result permits to transferring techniques and analytic concepts from the theory of *TCCS* to cIPA. Consider, for instance, two simple cIPA processes like $P = \text{rec}\ x.\ a.x$ and $Q = \text{rec}\ x.\ (a.x + a.x)$. By using the rules in Table 6, both P and Q give rise to two infinite transition systems thus, to prove the expected equivalence - $P \sim_{AM} Q$ - the standard algorithms for checking bisimulation-based equivalences cannot be used. $\Pi[\![P]\!]$ and $\Pi[\![Q]\!]$, instead, give rise to finite states transition systems (according to the rules in Table 3 and Table 4) and, hence, we could prove equivalence $\Pi[\![P]\!] \sim_{MT} \Pi[\![Q]\!]$ by exploiting the Concurrency Workbench [7], to deduce equivalence $P \sim_{AM} Q$. This reasoning can be generalized to the whole set of cIPA terms which are finite states according to the standard interleaving operational semantics (but, also in this sublanguage, the timed transition systems generated by the rules in Table 6 are usually infinite states because of the increasing of the local clocks values). We now mention to two further interesting results.

5.1 Dealing with Lazy Actions

Our results hold only when Well-Timed traces (the observation time coincides with the execution time) are taken into account from the cIPA operational semantics because in *TCCS* time always goes forward. They also hold only when processes that can stop the passage of time are removed by *TCCS*. This is because every process p which can cause a time stop, can consequently cause the deadlock of all processes which are in parallel composition or in alternative composition with p itself. This latter drawback can be overcome when dealing with the *lazy* versions of cIPA [6] and *TCCS* [19] where actions can delay their execution arbitrarily long. The central rule for Lazy cIPA is $n \Rightarrow a.P \xrightarrow[\Delta(a)]{a@n+t} n + t + \Delta(a) \Rightarrow P$ for every $t \in \mathbb{N}$ and similarly the central rule for Lazy *TCCS* is $a.p \overset{t}{\rightsquigarrow} a.p$ (and likely $\underline{0} \overset{t}{\rightsquigarrow} \underline{0}$) for every $t \in \mathbb{N}^+$. Every Lazy *TCCS* process can allow time to pass even if it is a (functionally) deadlocked process. This permits the whole Lazy cIPA (including restriction) to be encoded within Lazy *TCCS*.

5.2 Observing the Completing Time of Action Execution

One of the main assumptions behind the theory of processes with durational actions in [1,2], is that both the starting time of action execution and its duration are observable while experimenting over processes. A different approach is taken in [9,11,12] where, instead, only their completing time is taken into account; thus, Gorrieri and his co-authors, for instance, have transitions of the form $d \xrightarrow{\langle\mu,n\rangle} d'$ meaning that timed state d can become timed state d' by performing an action μ at completing time n. By following similar lines of Definition 6 we could define a well-timed operational semantics for cIpa based on the assumption that the completing time of action execution is observable. Hence, a strong bisimulation equivalence, $\sim^c_{AM}$, can be defined which (is unrelated with $\sim_{AM}$ and) agrees with Moller and Tofts's $\mathcal{T}$-equivalence; i.e. we can still prove that $P \sim^c_{AM} Q$ if and only if $\bar{\Pi}(Q) \sim_{MT} \bar{\Pi}(Q)$. The new mapping $\bar{\Pi}[\![\_]\!] : \mathcal{P}_{AM} \to \mathcal{P}_{MT}$ is different by that studied in definition 4.1. It is obtained by replacing in Definition 4.1, Π with $\bar{\Pi}$, and rules on the l.h.s of Table 7 with the corresponding ones on the r.h.s.

$\Pi[\![a.P]\!] = a.(\Delta(a)).\Pi[\![P]\!]$	$\bar{\Pi}(a.P) = (\Delta(a)).a.\bar{\Pi}(P)$
$\Pi[\![\text{wait } n.\, P]\!] = \tau.(n).\Pi[\![P]\!]$	$\bar{\Pi}(\text{wait } n.\, P) = (n).\tau.\bar{\Pi}(P)$

Table 7
New rules for action prefixings.

We conclude this section by mentioning to interesting lines of further work.

5.3 Compact Representations of Timed Transition Systems

Because of the Well-Timed character of the new cIpa operational semantics, a notion of *compact timed transition system* can be provided by following [6]. This notion allows to providing cIpa timed states (which are finite states according to the standard interleaving operational semantics) with a finite timed transition system, and timed bisimulation ($\sim_{AM}$) with a finite alternative characterization which makes this equivalence amenable to a mechanical treatment in the same vein as the classical bisimulation-based equivalences. For instance, processes $P = \text{rec } x.\ a.x$ and $Q = \text{rec } x.\ (a.x + a.x)$ would have compact transition systems $0 \Rightarrow \text{rec } x.\ a.x \xrightarrow[\Delta(a)]{a@0} 0 \Rightarrow \text{rec } x.\ a.x$ and $0 \Rightarrow \text{rec } x.\ (a.x + a.x) \xrightarrow[\Delta(a)]{a@0} 0 \Rightarrow \text{rec } x.\ (a.x + a.x)$ respectively. Note that these two transition systems are smaller (in terms of the number of states and transitions) than those associated to $\Pi[\![0 \Rightarrow \text{rec } x.a.x]\!]$ and $\Pi[\![0 \Rightarrow \text{rec } x.(a.x + a.x)]\!]$ (consider all possible $\overset{t}{\leadsto}$-transitions that the two latter processes can perform).

In general, we would like to prove that every (compact) transition out of a timed state d, represents a "symbolic" transition for a number of transitions modelling the passing of time out of $\Pi[\![d]\!]$. Thus, the classic partition refinement algorithms for checking the equivalence of compact timed states, would be faster than the corresponding ones for checking the equivalence of their translations.

5.4 Towards the Opposite Coding

Another interesting line of further work is that of studying codings $\Pi^r[\![\_]\!] : \mathcal{P}_{MT} \to \mathcal{P}_{AM}$ such that for all p, q in $\mathcal{P}_{MT}$ it is $p \sim_{MT} q$ if and only if $\Pi^r(p) \sim_{AM} \Pi^r(q)$ (if and only if the compact representations of $\Pi^r(p)$ and $\Pi^r(q)$ are related by $\sim_{AM}$). By this perspective, we have (i) to remove those processes which can cause time stops (processes that can stop the passage of time such as 0 and restriction) because in cIpa there are not such pathological processes and (ii) allows different durations (included null duration) to be associated to different occurrences of the same basic action (this can be done by using different prefixings, $a_n.\_$, like those in [9] where the duration n of an action a is explicitly reported in the action specification).

Acknowledgments

I would like to thank Frits Vaandrager who suggested me the possibility of relating the two timed processes algebras presented in this paper at the Express'95 meeting and Roberto Gorrieri for useful comments and suggestions.

References

[1] L.Aceto, D.Murphy: On the Ill–Timed but Well–Caused. In the Proceedings of CONCUR'93 (E.Best, Ed.), Springer-Verlag LNCS **715**, pp.97-111, 1993.

[2] L.Aceto, D.Murphy: Timing and Causality in Process Algebra. Acta Informatica **33** (4), pp.317-350, 1996.

[3] J.Baeten, J.Bergstra: Real Time Process Algebra. Formal Aspects of Computing **3** (2), pp.142-188, 1991.

[4] Bergstra,J.A., Klop,J.W.: Process Theory based on Bisimulation Semantics. In Linear Time, Branching Time and Partial Orders in Logic and Models for Concurrency, LNCS **354**, 1989.

[5] R.Cleaveland, V.Natarajan: An algebraic theory of process efficiency. In the Proceedings of LICS'96, pp.63-72, 1996.

[6] F.Corradini, M.Pistore: Specification and Verification of Timed Lazy Systems. In the Proceedings of MFCS'96, Springer-Verlag LNCS **1113**, pp.279-290, 1996.

[7] R.Cleaveland, J.Parrow, and B.Steffen: The concurrency workbench: A semantics-based tool for the verification of concurrent systems. In ACM Transaction on Programming Languages and Systems **15**, 1993.

[8] R.Cleaveland, A.Zwarico: A theory of testing for real-time. In the Proceedings of LICS'96, pp.110-119, 1991.

[9] G-L.Ferrari, U.Montanari: Dynamic matrices and the cost analysis of concurrent programs. In Proceedings of AMAST'95, LNCS **936**, pp.307-321, 1995.

[10] R. van Glabbeek, W.P. Weijland: Branching time and abstraction in bisimulation semantics. In Information Processing '89, pp.613-618, 1989.

[11] R.Gorrieri, M.Roccetti: Towards Performance Evaluation in Process Algebras. In the Proceedings of AMAST'93, Workshop in Computing Series, Springer-Verlag, pp.289-296, 1993.

[12] R.Gorrieri, M.Roccetti and E. Stancampiano: A Theory of Processes with Durational Actions. Theoretical Computer Science **140** (1), pp.73-94, 1995.

[13] Hoare, C.A.R.: *Communicating Sequential Processes.* Prentice Hall 1989.

[14] M.Hennessy, T.Regan: A temporal process algebras. Information and Computation **117**, pp.221-239, 1995.

[15] S.Arun-Kumar, M.Hennessy: An Efficiency Preorder for Processes. Acta Informatica **29**, pp.737-760, 1992.

[16] R.Milner: *Communication and concurrency.* International series on computer science, Prentice Hall International, 1989.

[17] Milner,R., Parrow,J., Walker,D.: A Calculus of Mobile Processes (I and II). Information and Computation **100** (1), pp. 1-77, 1992.

[18] F.Moller, C.Tofts: A Temporal Calculus of Communicating Systems. In the Proceedings of CONCUR'90, LNCS **459**, Springer-Verlag, pp.401-415, 1990.

[19] F.Moller, C.Tofts: Relating Processes with Respect to Speed. In the Proceedings of CONCUR'91, LNCS **527**, Springer-Verlag, pp.424-438, 1991.

[20] G.M.Reed, A.W.Roscoe: A timed model for communicating sequential processes. Theoretical Computer Science **58**, pp.249-261, 1988.

[21] W.Vogler: Timed Testing of Concurrent Systems. Information and Computation **121**(2), pp.149-171, 1995.

[22] W.Yi: Real time behaviour of asynchronous agents. In the Proceedings of CONCUR'90, LNCS **458**, Springer-Verlag, pp.502-520, 1990.